Bruce Xi

(236)-777-8218 • brucexi99@outlook.com • bruce-shidi-xi • brucexi999

Education

The University of British Columbia **Vancouver, BC**

Master of Engineering in Electrical and Computer Engineering, GPA 4.0 *2021–Present*

- Relevant courses: Deep Learning, Reinforcement Learning, ML Hardware Accelerator, Computer Architecture, Digital/Microcomputer System Design, VLSI, IC Testing and Reliability

Imperial College London **London, UK**

Bachelor of Engineering in Materials Science and Engineering *2018–2021*

- Graduated with First-Class Honours
- Obtained Dean's List for three consecutive years (2018-2021)

Experience

Motorola Solutions **Vancouver, BC**

Design Validation Co-op *May-Dec. 2022*

- Conducted extensive surveillance camera tests in various settings, validating the cameras' electrical, mechanical, and optical performance
- Engineered Python-based software for test automation and data analysis, enhancing test efficiency with up to 90% automation
- Utilized FFmpeg for video processing and analysis
- Collaborated within a team using Git, Confluence, and Jira to optimize workflow efficiency

Project

Concurrent VLSI Routing with Multi-agent Deep Reinforcement Learning **May-Oct. 2023**

- Developed a Python-based machine learning framework to address the VLSI global routing problem in a concurrent manner. This framework modeled routing as a pathfinding task and solved it using multi-agent reinforcement learning integrated with deep neural networks (MLP and GNN implemented in PyTorch)
- Addressed training challenges by fine-tuning hyperparameters through a grid search approach, leading to significant performance improvements
- The proposed work overcame the traditional net-ordering issue, guaranteed zero overflow, and outperformed an A* baseline by 2.6% in terms of wirelength

Computer Architecture Simulator **Sept.-Dec. 2023**

- Leveraged tiling and loop reordering to optimize matrix multiplication in C. These techniques changed memory access patterns and improved cache hit rate by 8x
- Developed four cache replacement policies in C++ for the ChampSim simulator, achieving Instruction Per Cycle (IPC) and hit rates comparable to the Least Recently Used (LRU) policy
- Implemented two branch prediction algorithms in C++, resulting in IPC and accuracies on par with an established baseline

Microcomputer System Development **Jan.-Apr. 2023**

- Engineered key components of a microcomputer system including a 4-way set-associative cache and a DRAM controller using Verilog. The cache reduced the runtime of a benchmark by 43%
- Implemented the system on an Altera FPGA with a soft core provided by the course
- Developed software and firmware in C that interacted with hardware (Flash, EEPROM, and ADC/DAC) using SPI, IIC, and CAN protocol
- Utilized hardware timer interrupt and designed a snake game software that ran on the system and interact with the player using VGA

CPU Design and Assembly programming **June-Sept. 2022**

- Architected a 16-bit RISC CPU from the ground up using Verilog, integrating pivotal components such as FSM, datapath, RAM, and I/O interfaces. The CPU was implemented on an Altera FPGA
- The CPU supported 13 diverse instructions encompassing ALU operations, memory access, and branching mechanisms
- Implemented preemptive multitasking on an ARM core integrated on the FPGA using Assembly

Skills

Hardware: Verilog, FPGA, ModelSim, Quartus, Cadence

Software: Python > C > ARM Assembly = C++, Linux

DevOps Tools: Confluence, Jira, Git, GitHub